

基本原则

- 尽快呈现 UI 给用户，保持交互过程流畅
- 减轻服务器压力，减少网络流量

减少 HTTP 请求数

- 资源合并与压缩
- CSS Sprites
- 小图片使用 base64 做 Inline Image
- 图片懒加载
- 按需加载 js
- 避免请求重复资源, 尽量缓存请求结果

减少页面重绘和脚本阻塞

- 将 CSS 放在 head 中
- 将 js 置于 body 底部
- 异步加载 js

图片优化

- 所有图片必须经过压缩
- 单张图片尽量保持在 300KB 以下
- 矢量风格图片使用 .svg
- 色彩较少的图片使用 .png , 色彩较丰富的图片使用 .jpg
- 不需 alpha 透明的图片可以使用 png8 保存

基本原则

结构、样式、行为分离

尽量保持模板只包含 html 结构，样式放在 css 里，行为放在 js 里。

文件编码

使用不带 BOM 的 UTF-8 编码

- 在 html 指定 `<meta charset="utf-8">`
- css 中**不需要**指定 `@charset`

缩进

统一缩进格式为4个空格

一律使用小写字母

```
<!-- Recommended -->


<!-- Not recommended -->
<A HREF="/">Home</A>
```

```
/* Recommended */
color: #e5e5e5;

/* Not recommended */
color: #E5E5E5;
```

引号

在 html 和 css 中使用双引号, js 中使用单引号

```
<!-- Not recommended -->
<span id='j-hook' class=text>Google</span>

<!-- Recommended -->
<span id="j-hook" class="text">Google</span>
```

```
/* Not recommended */
body {
  font-family: 'open sans', arial, sans-serif;
  background: url(..img/bg.jpg);
}
.selector[type=text] {...}

/* Recommended */
body {
  font-family: "open sans", arial, sans-serif;
  background: url("../img/bg.jpg");
}
.selector[type="text"] {...}
```

```
/* Not recommended */
var html = "<div class=\"logo\">Google</div>";

/* Recommended */
var html = '<div class="logo">Google</div>';
```

代码验证

- 使用 (W3C HTML Validator)[http://validator.w3.org/] 来验证HTML 代码有效性
- 使用 (W3C CSS Validator)[http://jigsaw.w3.org/css-validator/validator.html.zh-cn] 来验证CSS代码有效性

HTML

标签

- 自闭合（self-closing）标签，无需闭合 (例如： `img input` 等);
- 可选的闭合标签（closing tag）, 一定要闭合 (例如： ``);
- 尽量减少标签数量;

```

<input type="text" name="title">

<ul>
  <li>Style</li>
  <li>Guide</li>
</ul>
```

属性

- 属性 (如 `class` 、 `id`) 单词字母小写，多个单词组成时，采用中划线 `-` 分隔
- 自定义属性使用 `data-` 开头

```
<!-- Not recommended -->
<div class="contentWrapper" slider="on"></div>

<!-- Recommended -->
<div class="content-wrapper" data-slider="on"></div>
```

布尔值属性

HTML5 规范中 `disabled` 、 `checked` 、 `selected` 等属性不用设置值。

```
<input type="checkbox" value="1" checked>
```

基本

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Style Guide</title>
</head>
```

CSS

不要使用 @import

- 合并 css 文件
- 使用 `<link>` 代替

选择器

- 使用连字符 `-` 作为选择符名称界定符，不要驼峰命名法和下划线
- 避免选择器嵌套层级过多，尽量少于 3 级
- 避免使用通用选择器 `*`
- 避免不必要的选择器

```
/* Not recommended */
.content * {color: red;}
button#back-button {...}
.menu ul li {...}

/* Recommended */
#back-button {...}
.menu li {...}
```

换行

```
/* Not recommended */
.selector, .selector-secondary, .selector[type=text] {
  padding:15px;
  margin:0px 0px 15px;
}

/* Recommended */
.selector,
.selector-secondary,
.selector[type="text"] {
  padding: 15px;
  margin-bottom: 15px;
}
```

媒体查询（Media query）的位置

`@media` 放在尽可能相关规则的附近，避免被遗忘

```
.element { ... }
.element-avatar { ... }
.element-selected { ... }

@media (max-width: 768px) {
  .element { ...}
  .element-avatar { ... }
  .element-selected { ... }
}
```

JavaScript

全局变量是魔鬼

- 尽量减少全局变量
- 使用命名空间来控制全局变量
- 使用立即执行函数来避免全局变量

```
window.global = {
  siteName: 'google',
  host: {
    server: 'api.google.com',
    cdn: 'cdn.google.com'
  },
  brower: { /* ... */ }
}

(function(exports){
  var ua = navigator.userAgent;
  if (/android|htc/.test(ua)) {
    exports.isAndroid = true
  }
  // ...
})(global)
```

命名规则

- 变量/方法/属性/参数, 使用 Camel 命名
- 私有属性/变量/方法以下划线 `_` 开头
- 类/枚举变量, 使用 Pascal 命名法
- 常量/枚举的属性, 使用全部字母大写，单词间下划线 `_` 分隔的命名方式
- 尽量不用简写，可读性强

```
var loadingModules = {};  
var _privateMethod = {};  
var HTML_ENTITY = {};  
function Book(name) {  
    this.name = name;  
}  
var TargetState = {  
    READING: 1,  
    READED: 2,  
    APPLIED: 3,  
    READY: 4  
};
```

Copyright © 2015, 平安金管家

Documentation built with [MkDocs](#).